



PROGETTO INGEGNERIA DEL SOFTWARE

Parte 2 - UML

Abiuso Matteo

Barbagiovanni Minciullo Carlo

Consonni Luca



SOMMARIO

○ Diagrammi comportamentali:

- Use case diagram: Evento
- Activity diagram: Iterazione manager-cliente, Calcolare guadagno
- Sequence diagram: Funzioni calcolo fatturato, Registrazione progetto
- Statechart diagram: Stato dell'ordine, Schermata inserimento dati
- Communication diagram: Registrazione progetto

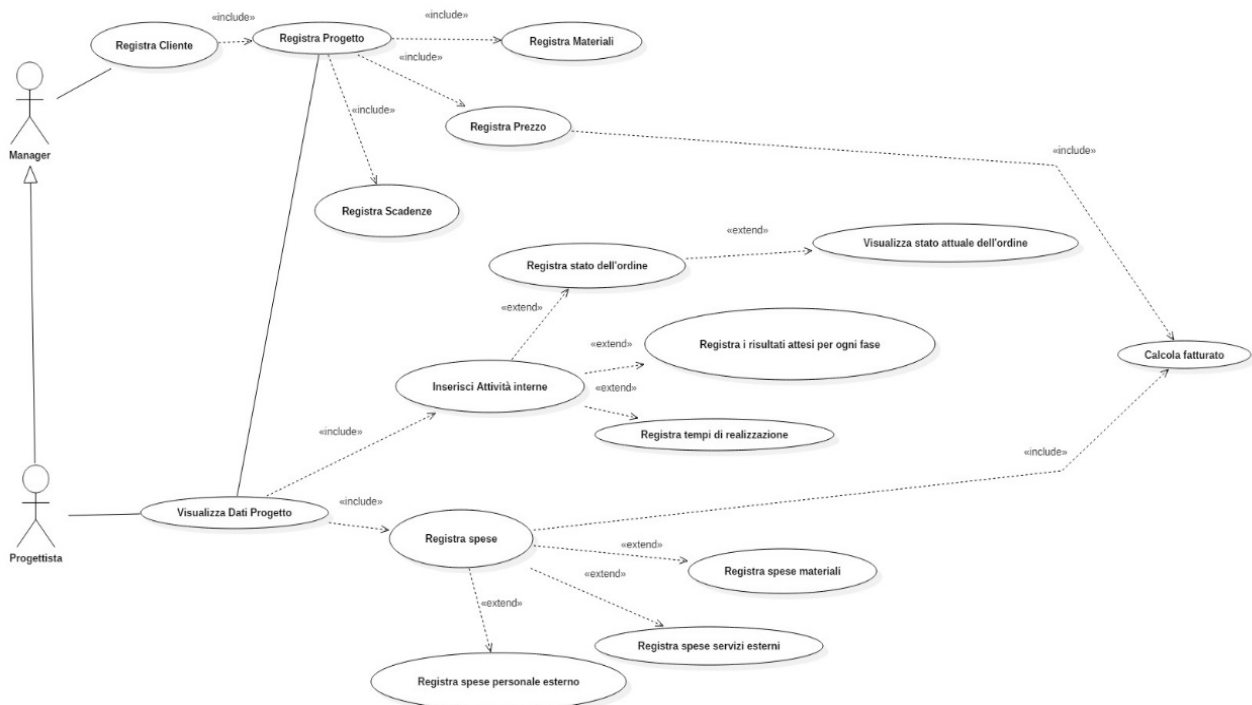
○ Diagrammi strutturali:

- Class Diagram
- Object Diagram: Evento
- Component Diagram
- Deployment Diagram

DIAGRAMMI COMPORTAMENTALI

USE CASE DIAGRAM

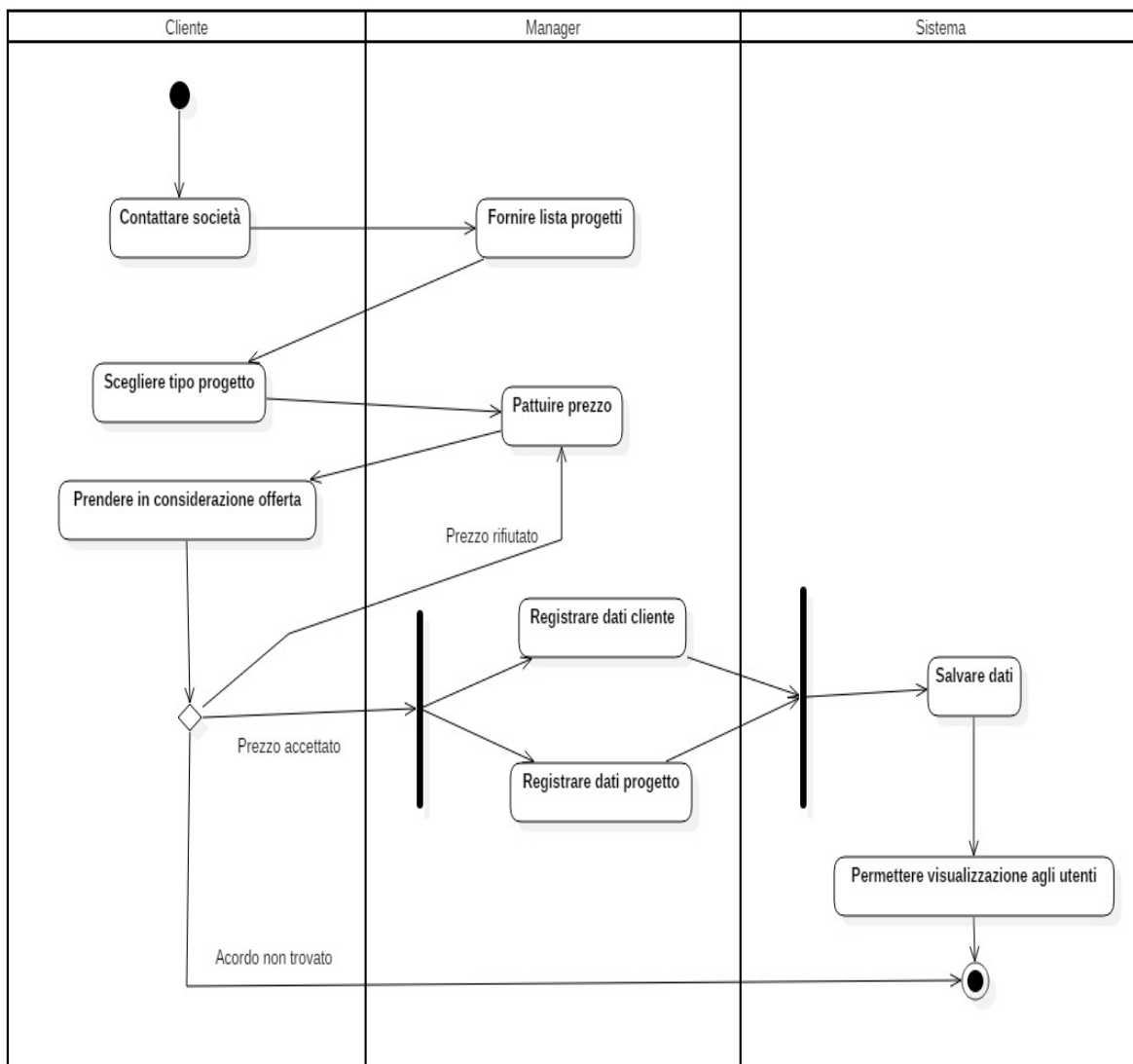
EVENTO



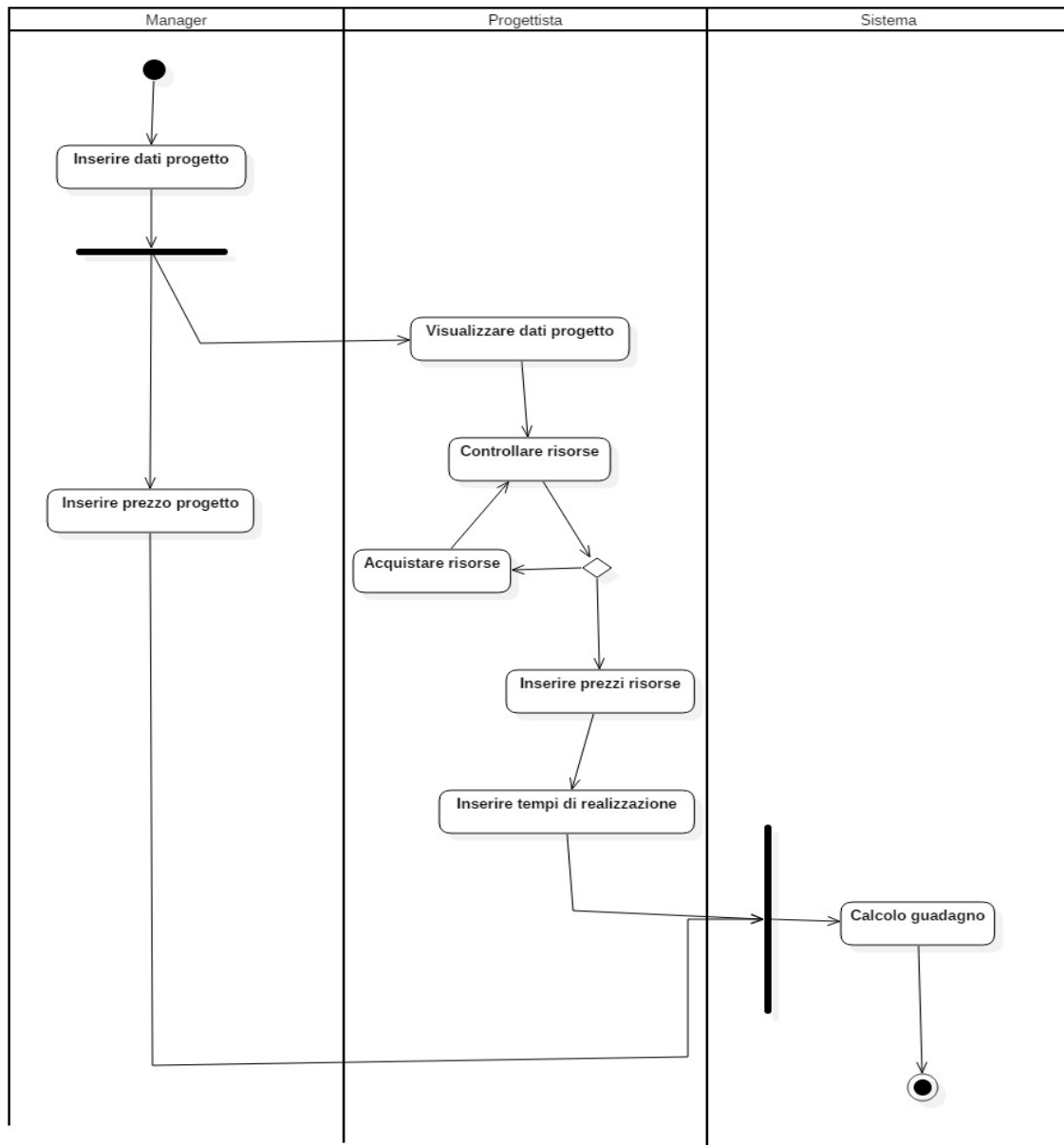
Lo **Use Case** descrive le funzioni e i servizi offerti dal sistema, così come sono percepiti e utilizzati dagli attori che interagiscono con il sistema stesso. In questo caso particolare abbiamo deciso di analizzare le iterazioni sistema-attori per la creazione e realizzazione di un “Evento”.

ACTIVITY DIAGRAM

ITERAZIONE MANAGER-CLIENTE



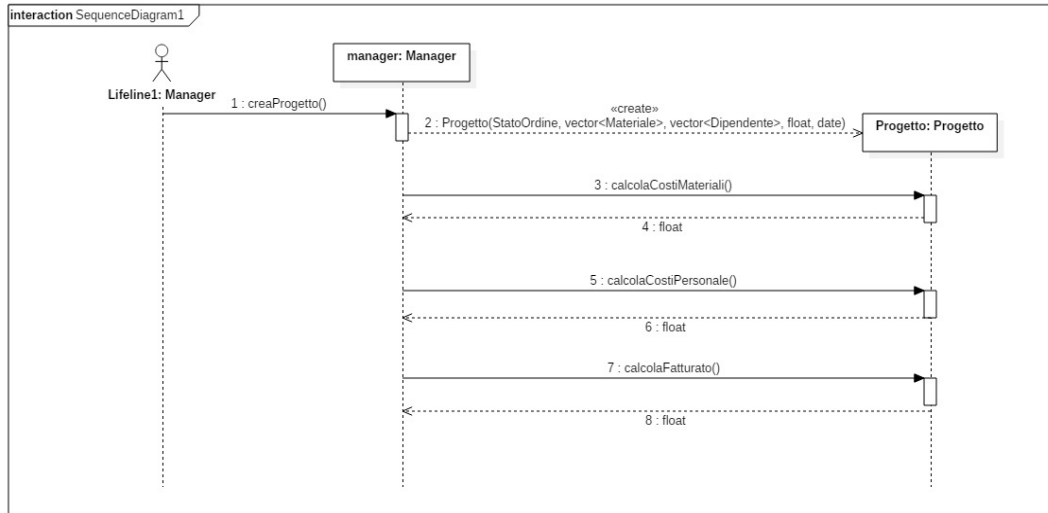
CALCOLARE GUADAGNO



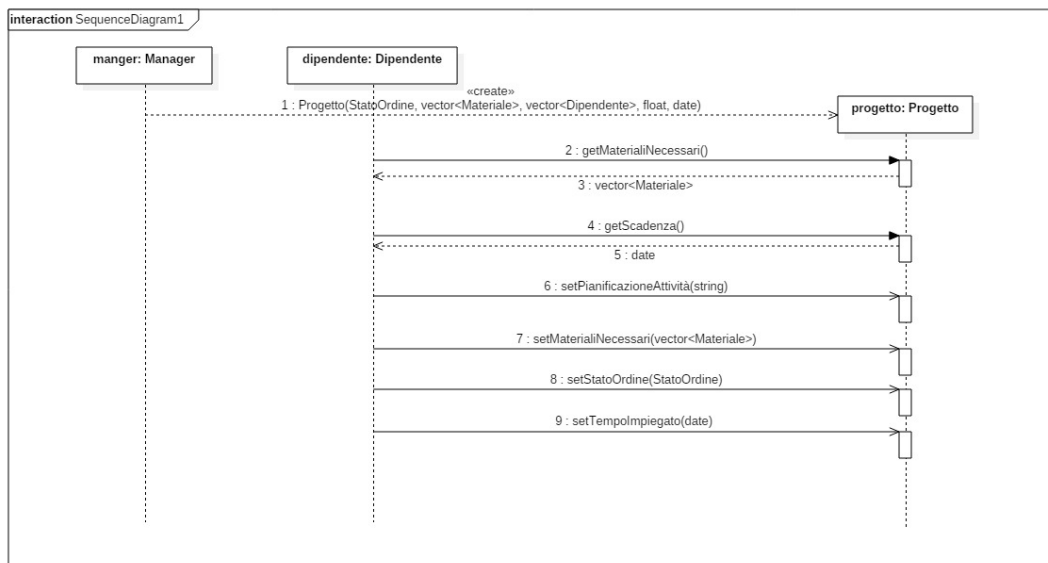
L'**Activity Diagram** definisce le attività da svolgere per realizzare una funzionalità, in questo caso: iterazione manager-cliente e calcolo guadagno. Più in dettaglio, un *activity diagram* definisce una serie di attività o flusso, anche in termini di relazioni tra le attività, i responsabili per le singole attività e i punti di decisione.

SEQUENCE DIAGRAM

CALCOLO FATTURATO



REGISTRAZIONE PROGETTO

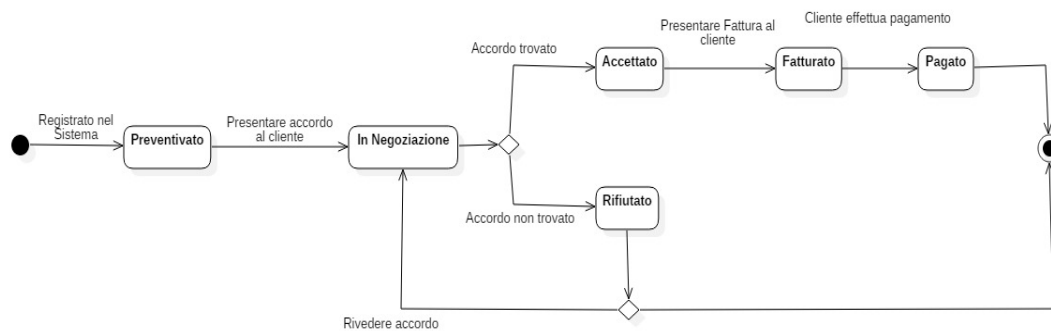


Usiamo il **Sequence Diagram** per rappresentare due scenari, ovvero una determinata sequenza di azioni in cui le scelte sono già state effettuate, non compaiono né scelte né flussi alternativi. In particolare abbiamo deciso di descrivere le funzioni per il calcolo del fatturato e per la registrazione base di un progetto. Abbiamo messaggi sincroni quando l'emittente

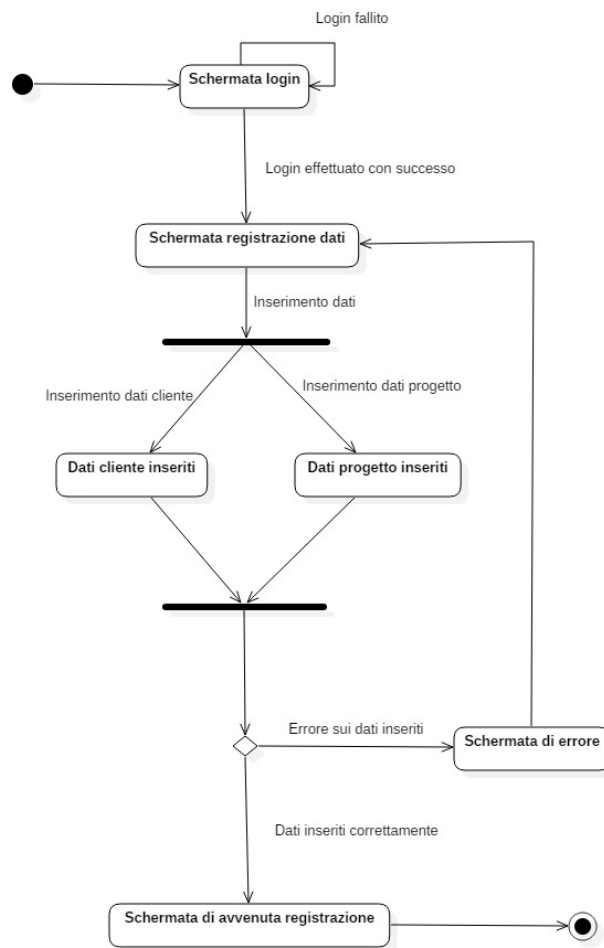
prima di procedere deve attendere una risposta, mentre asincroni, nel caso contrario.

STATECHART DIAGRAM

STATO DELL'ORDINE



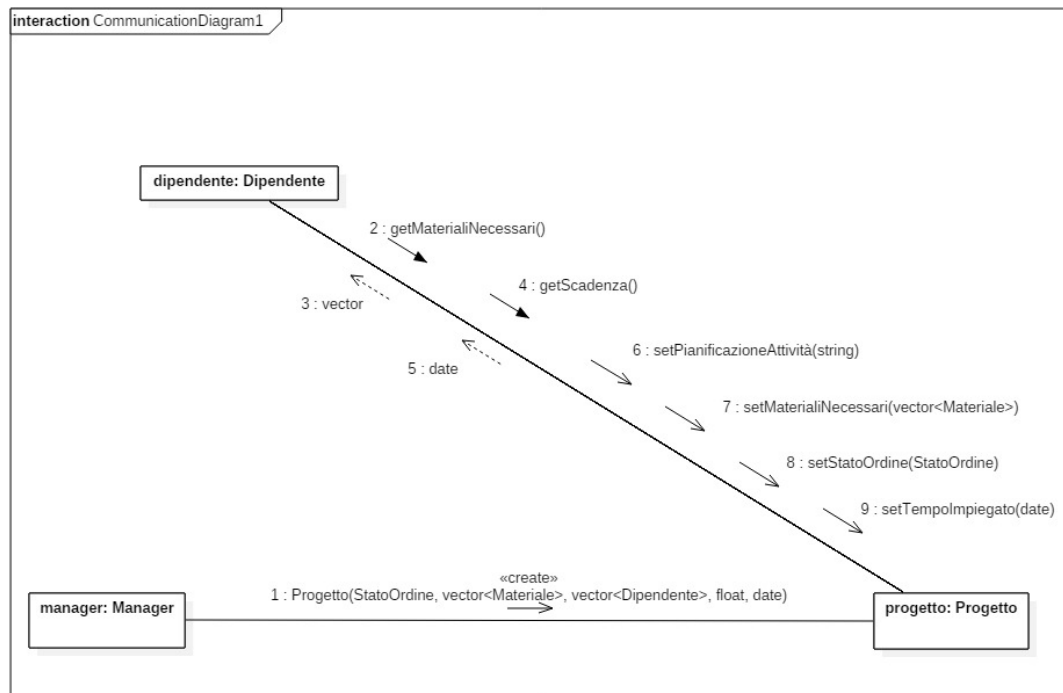
SCHERMATA DI LOGIN



Lo **Statechart Diagram** è un diagramma previsto dall'UML per descrivere il comportamento di entità o di classi in termini di stato.

Il diagramma mostra gli stati che sono assunti dallo stato dell'ordine e dalla schermata di login in risposta ad eventi esterni.

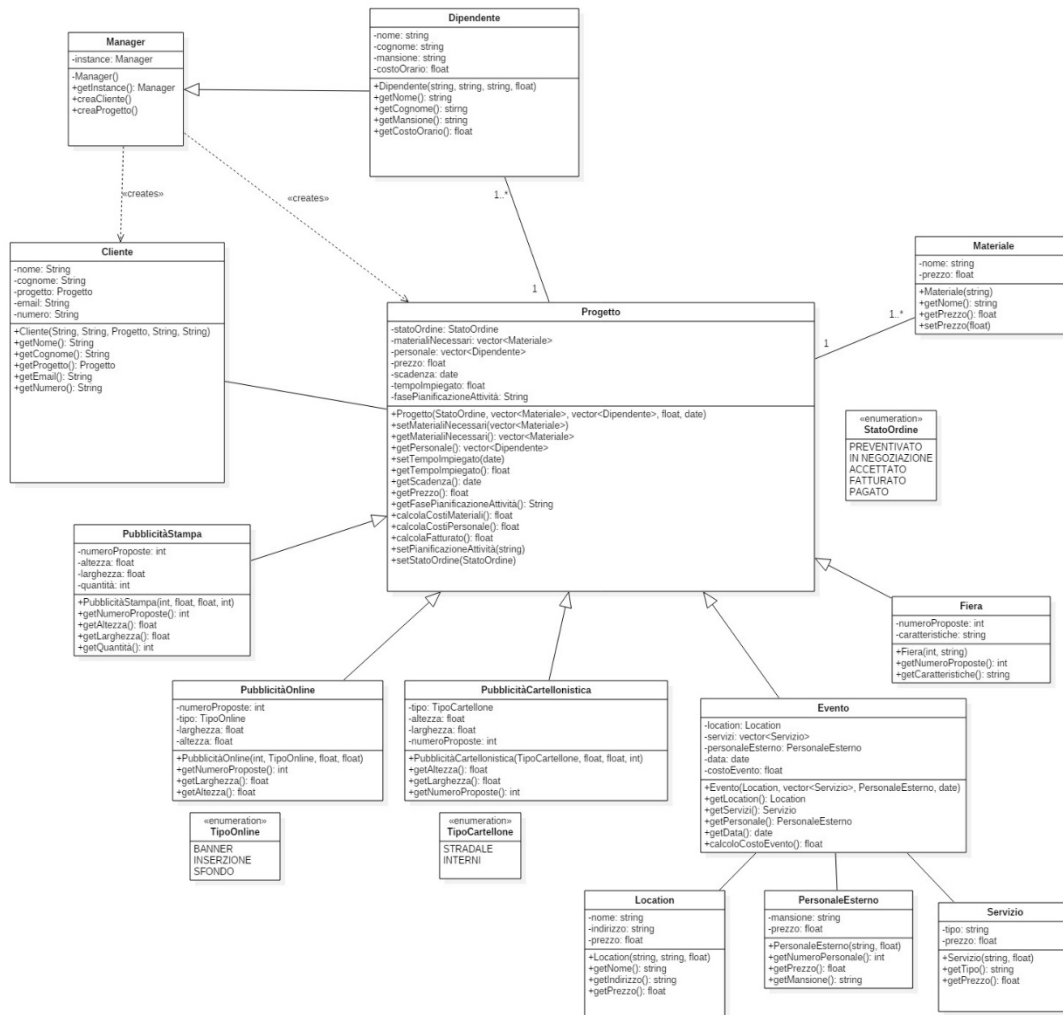
COMMUNICATION DIAGRAM



Un **Communication Diagram** è necessario per descrivere l'interazione fra più partecipanti alla realizzazione di una certa funzionalità. Insieme al Sequence Diagram fa parte dei cosiddetti diagrammi di interazione (*Interaction diagrams*), che sono spesso utilizzati per specificare come collaborano i vari componenti del software. Per questo motivo abbiamo scelto di descriverne la medesima iterazione (la registrazione base del progetto).

DIAGRAMMI STRUTTURALI

CLASS DIAGRAM



Class Diagram: Una classe rappresenta una categoria di entità (istanze), nel caso particolare dette oggetti; il nome della classe indica la categoria di entità descritta dalla classe. Ogni classe è corredata da un insieme di attributi (che descrivono le caratteristiche o lo stato degli oggetti della classe) e operazioni (che descrivono il comportamento della classe). Due classi possono essere legate da relazioni che rappresentano i legami (link) che possono sussistere fra gli oggetti delle classi associate. Esistono diversi tipi di relazione (associazione,

aggregazione, composizione, dipendenza, generalizzazione, realizzazione. Due classi possono essere legate da:

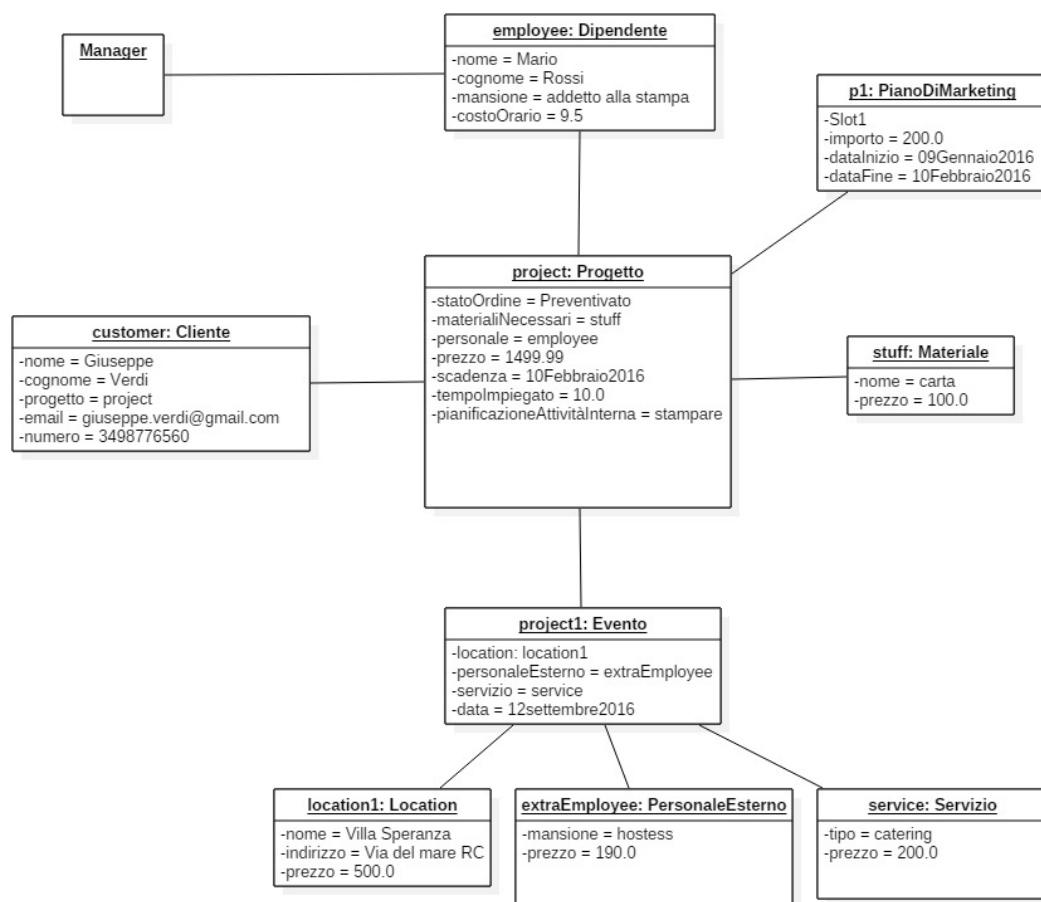
- relazione di associazione, che indica che gli oggetti delle due classi condividono una relazione statica.

- relazione di dipendenza, che indica che la definizione di una delle due fa riferimento alla definizione dell'altra.

- relazione di generalizzazione, che indica che una delle due classi (detta superclasse) si può considerare una generalizzazione dell'altra (detta sottoclasse).

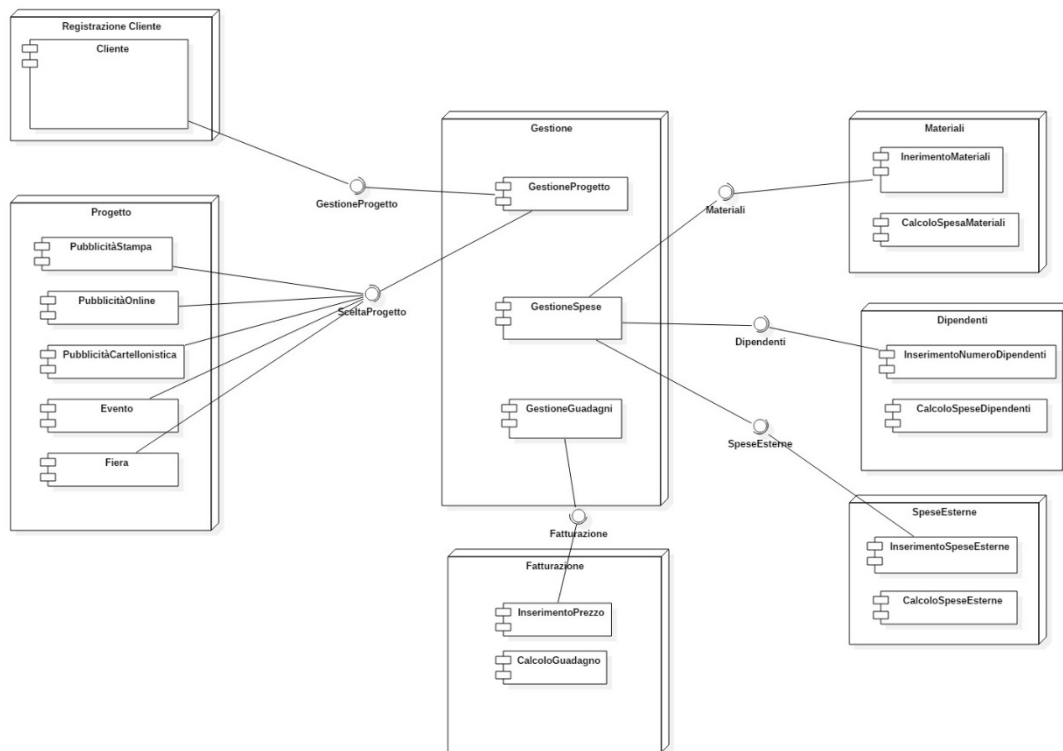
OBJECT DIAGRAM

EVENTO



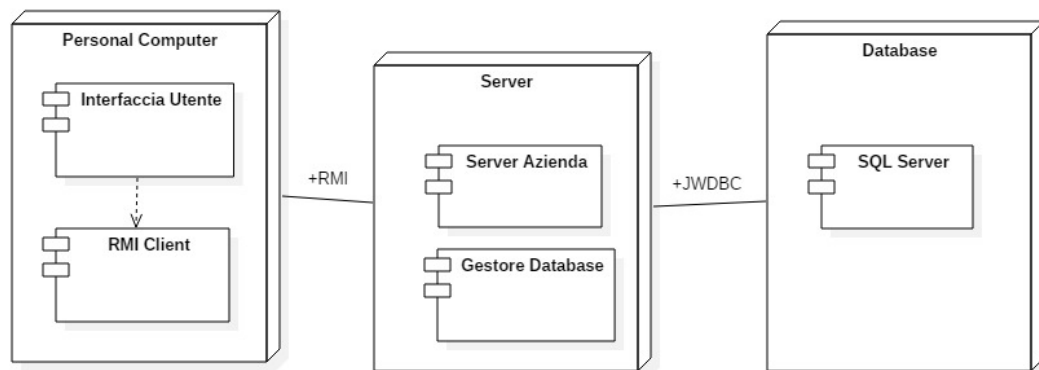
L'**Object Diagram** è una esemplificazione del class diagram. Nel particolare abbiamo descritto come si presenterebbero le classi nel caso con il nostro sistema dovessimo registrare un progetto evento.

COMPONENT DIAGRAM



Il Component diagram è un diagramma che ha lo scopo di rappresentare la struttura interna del sistema software modellato in termini dei suoi componenti principali e delle relazioni fra di essi.

DEPLOYMENT DIAGRAM



Il **Deployment Diagram** è un diagramma statico che descrive un sistema in termini di risorse Hardware, dette nodi, e di relazioni tra di esse.